What Do Hebbian Learners Learn?

Reduction Axioms for Iterated Hebbian Learning

Caleb Schultz Kisby,

with Saúl Blanco, Larry Moss

Indiana University

AAAI 2024

February 22, 2024

Foundations for Neuro-Symbolic AI

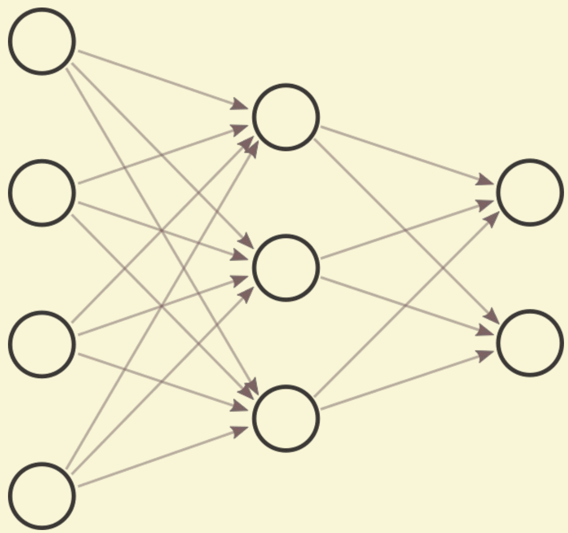
From van Harmelen (2022):

> "What are the possible interactions between knowledge and learning? Can reasoning be used as a symbolic prior for learning . . . Can symbolic constraints be enforced on data-driven systems to make them safer? Or less biased? Or can, vice versa, learning be used to yield symbolic knowledge? And if so, how to manage the inherent uncertainty that comes with such learned knowledge . . ."
>
> ". . . **neuro-symbolic systems currently lack a theory that even begins to ask these questions, let alone answer them.**"

van Harmelen, F. "Preface: The 3rd AI Wave Is Coming, and It Needs a Theory". In: Neuro-Symbolic Artificial Intelligence. Ed. by P. Hitzler and M. Sarker. IOS Press BV, 2022.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



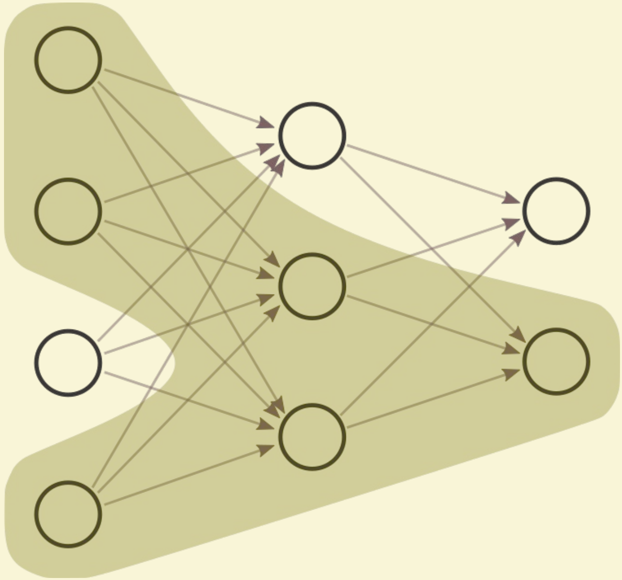
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



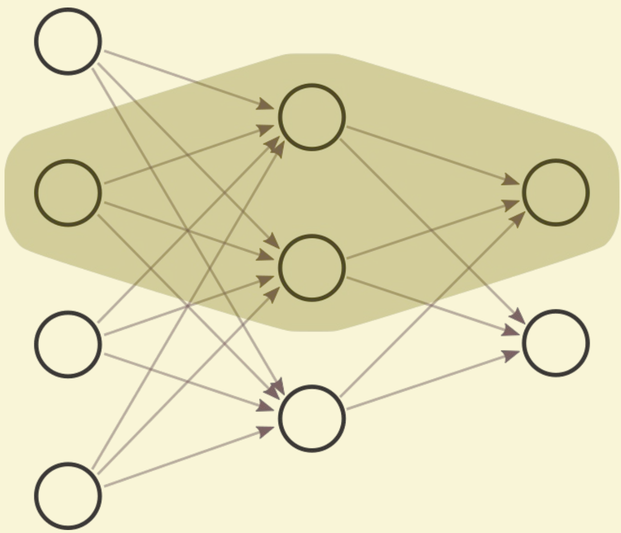
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



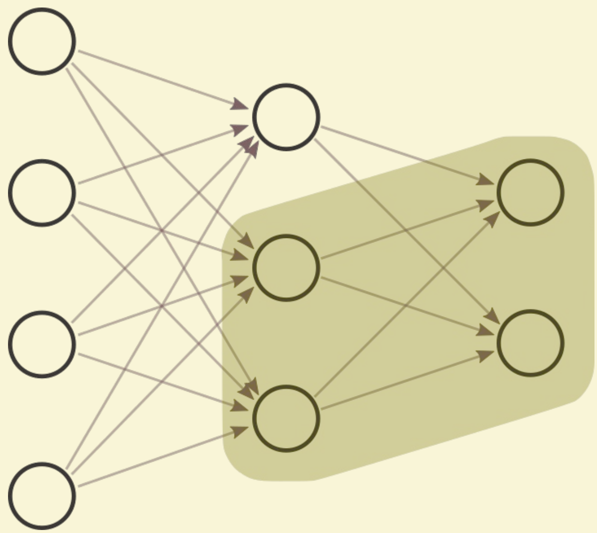
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



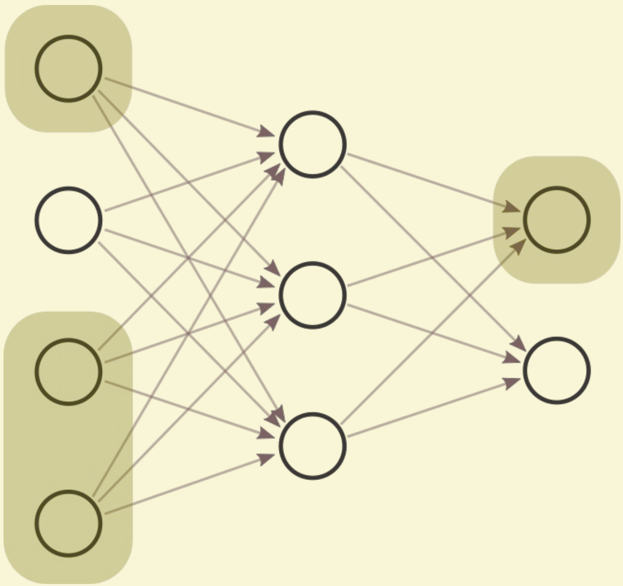
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



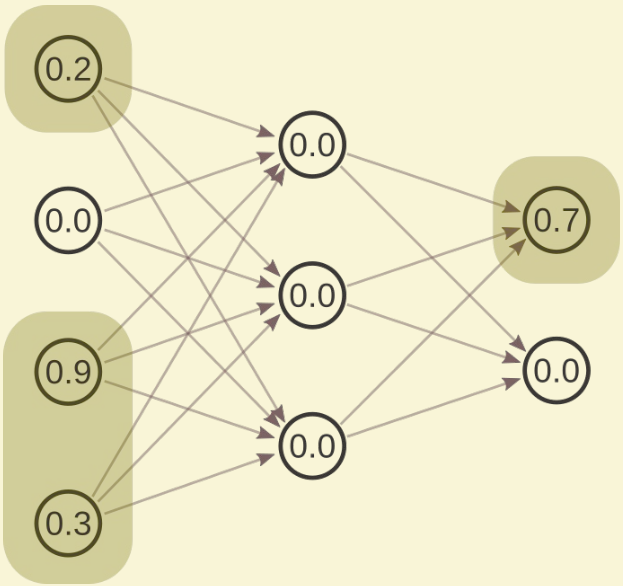
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



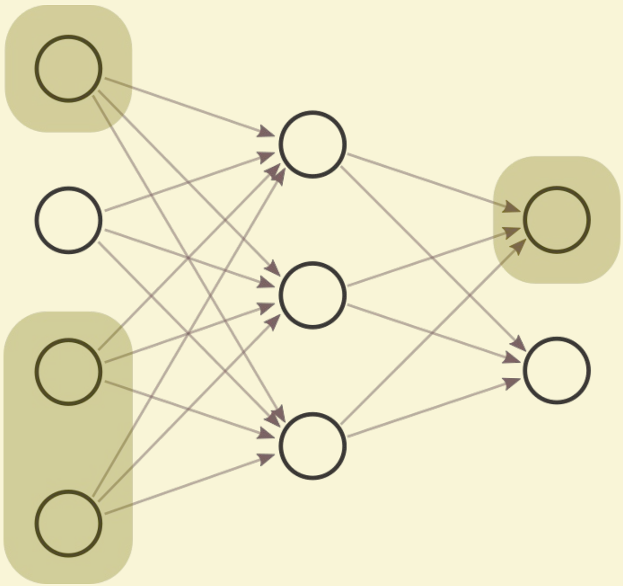
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics

- **We assume:** The network is weighted, feed-forward, fully-connected, with binary activations. The net' s states (activation patterns) are just given by sets of nodes.



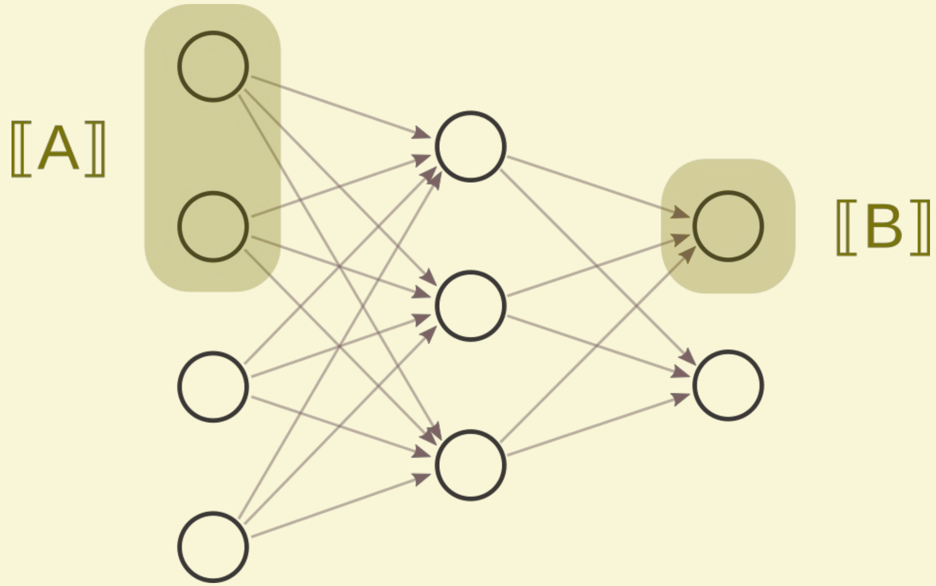
- **Key Idea:** Neural networks are not merely black boxes! Instead, think of nets as a kind of (logical) model; The dynamics of its states contain information about its conditional beliefs.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics (Contd.)

- An input state will activate new nodes, which subsequently activate more nodes. The forward propagation **Prop(S)** is the set of all neurons that are eventually activated by S.



The net satisfies A ⇒ B iff Prop(⟦A⟧) ⊇ ⟦B⟧

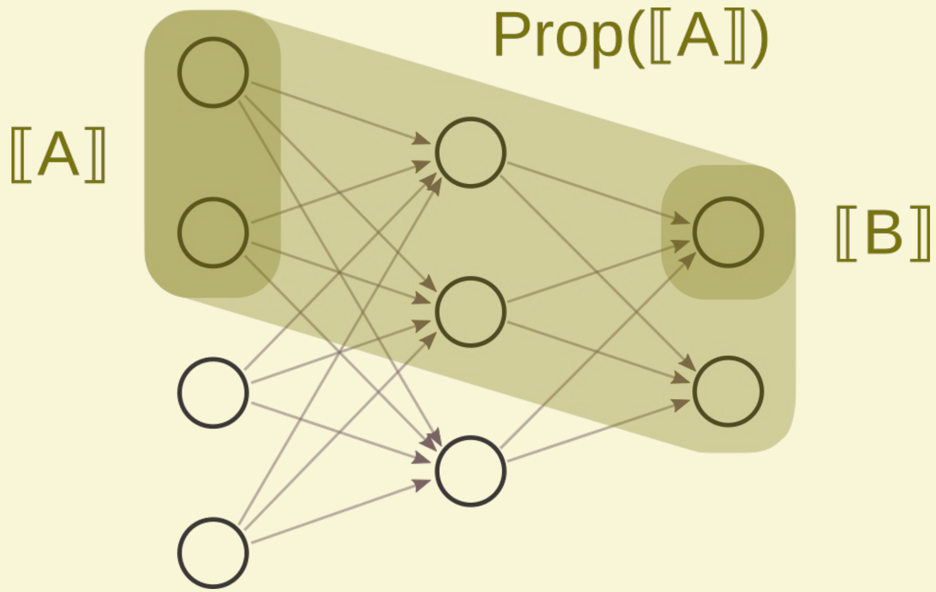
In other words, the net *classifies A as B*.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Neural Network Semantics (Contd.)

- An input state will activate new nodes, which subsequently activate more nodes. The forward propagation **Prop(S)** is the set of all neurons that are eventually activated by S.



The net satisfies A ⇒ B iff Prop(⟦A⟧) ⊇ ⟦B⟧

In other words, the net *classifies A as B*.

Balkenius, C. and Gardenfors, P. Nonmonotonic inferences in neural networks. In KR, 32–39. Morgan Kaufmann, 1991.

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Example: Building a Neural Network

Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

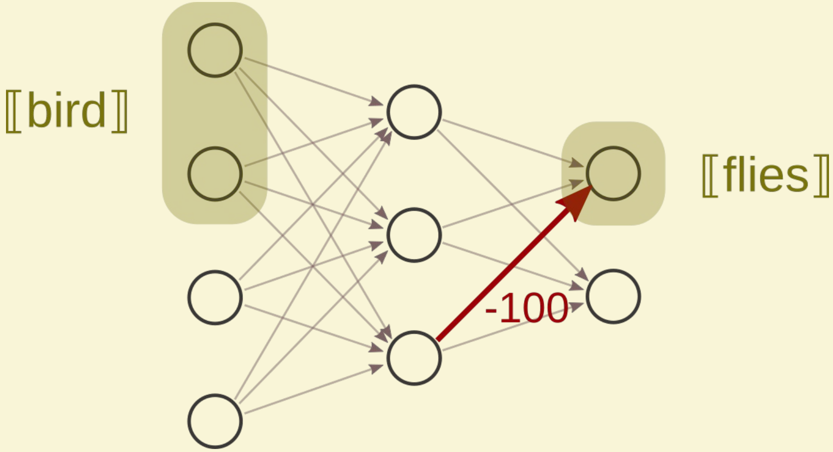
$$\text{penguin} \rightarrow \text{bird}$$
$$\text{bird} \Rightarrow \text{flies}$$
$$\neg\,(\text{penguin} \Rightarrow \text{flies})$$

Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

penguin $\rightarrow$ bird
bird $\Rightarrow$ flies
$\neg$ (penguin $\Rightarrow$ flies)

$\rightsquigarrow$

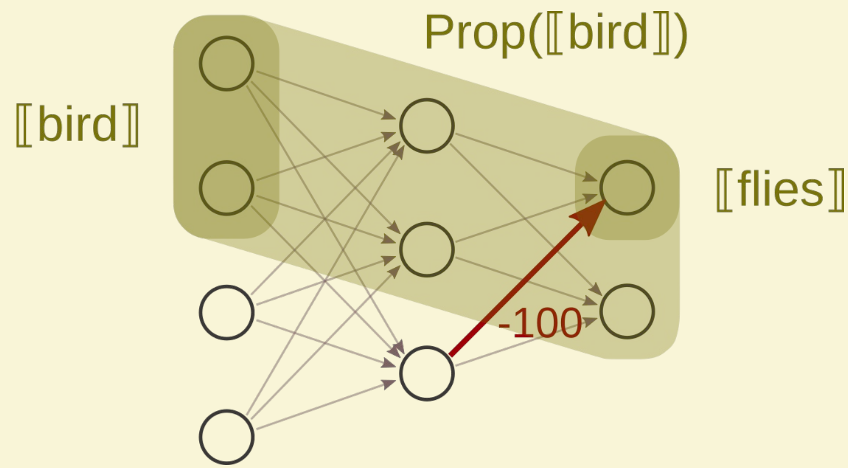


Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

$\text{penguin} \rightarrow \text{bird}$
$\text{bird} \Rightarrow \text{flies}$
$\neg(\text{penguin} \Rightarrow \text{flies})$

$\rightsquigarrow$

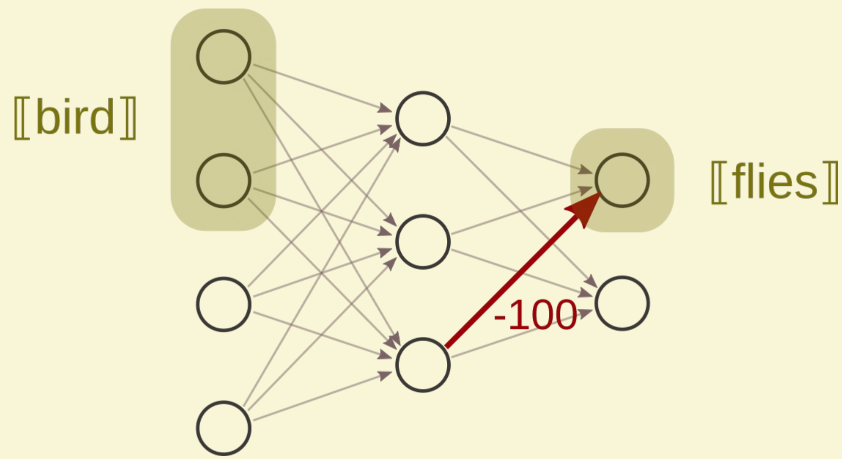


Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

$$\text{penguin} \rightarrow \text{bird}$$
$$\text{bird} \Rightarrow \text{flies}$$
$$\neg\,(\text{penguin} \Rightarrow \text{flies})$$

$\rightsquigarrow$

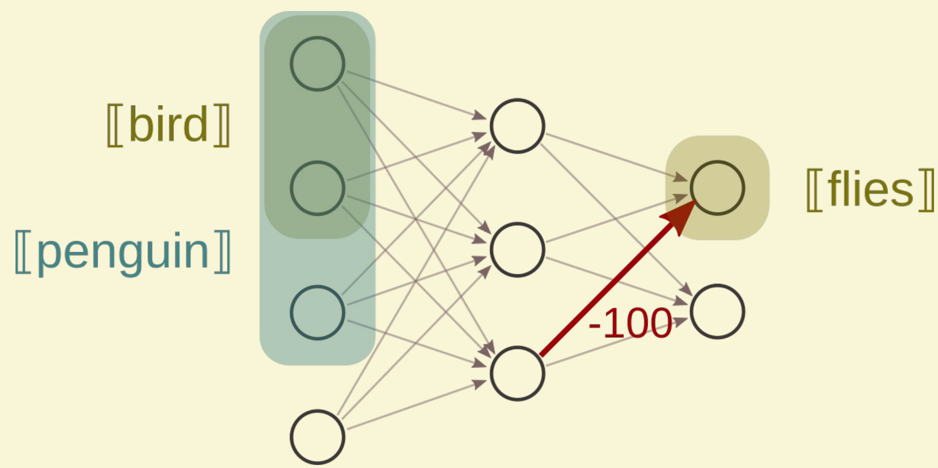


Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

$\text{penguin} \rightarrow \text{bird}$
$\text{bird} \Rightarrow \text{flies}$
$\neg\,(\text{penguin} \Rightarrow \text{flies})$

$\rightsquigarrow$

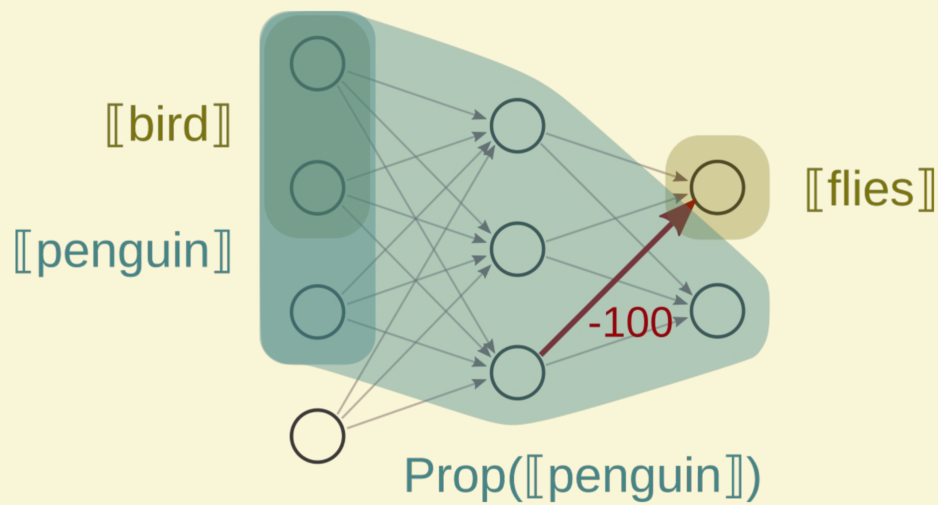


Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Example: Building a Neural Network

penguin → bird
bird ⇒ flies
¬ (penguin ⇒ flies)

⇝



Louis Agassiz Fuertes. Penguins (1895). Watercolor and pencil on paper. Image from Cornell University Lab of Ornithology Art Collection.

Soundness and Completeness

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Soundness and Completeness

Soundness

$\Gamma \vdash A$ implies $\Gamma \models A$

- **Not:** An explanation of a *particular* neural network's behavior
- **But instead:** Sound rules give *high-level* properties for *all* neural networks (of a certain architecture)

Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Soundness and Completeness

Soundness

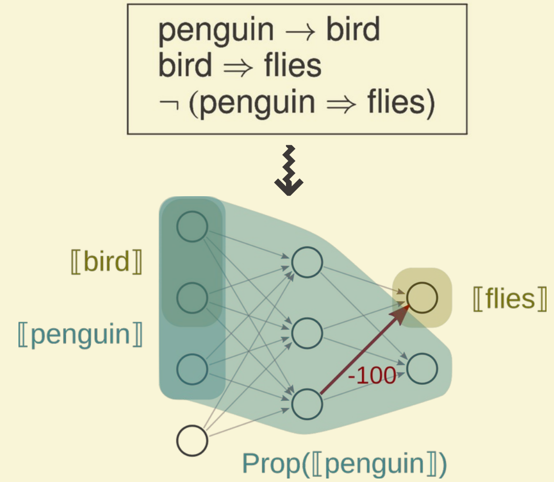
$\Gamma \vdash A$ implies $\Gamma \models A$

- **Not:** An explanation of a *particular* neural network's behavior
- **But instead:** Sound rules give *high-level* properties for *all* neural networks (of a certain architecture)

Completeness

$\Gamma \models A$ implies $\Gamma \vdash A$

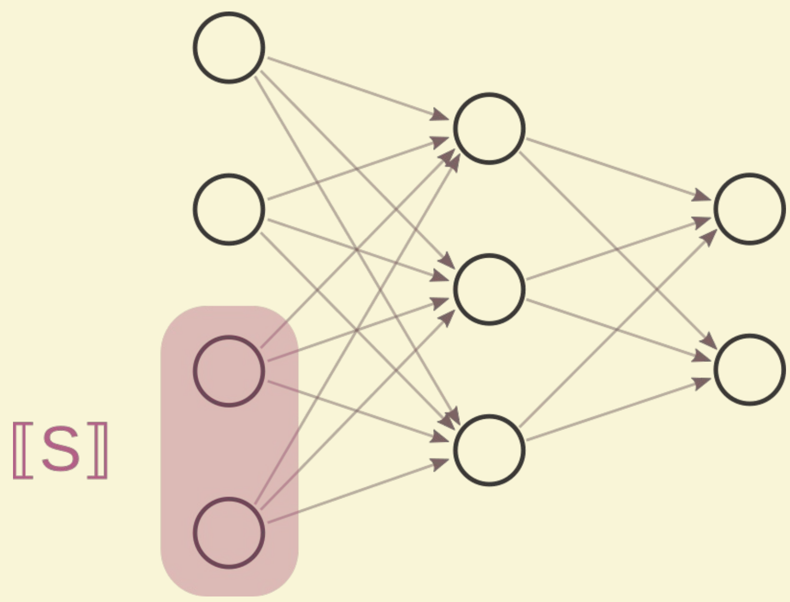
- **Equivalently:** Can we build a neural network satisfying the set Γ of constraints?



Leitgeb, H. Neural Network Models of Conditionals. In Introduction to Formal Philosophy, 147–176. Springer, 2018.

Iterated Hebbian Learning

Neurons that fire together wire together



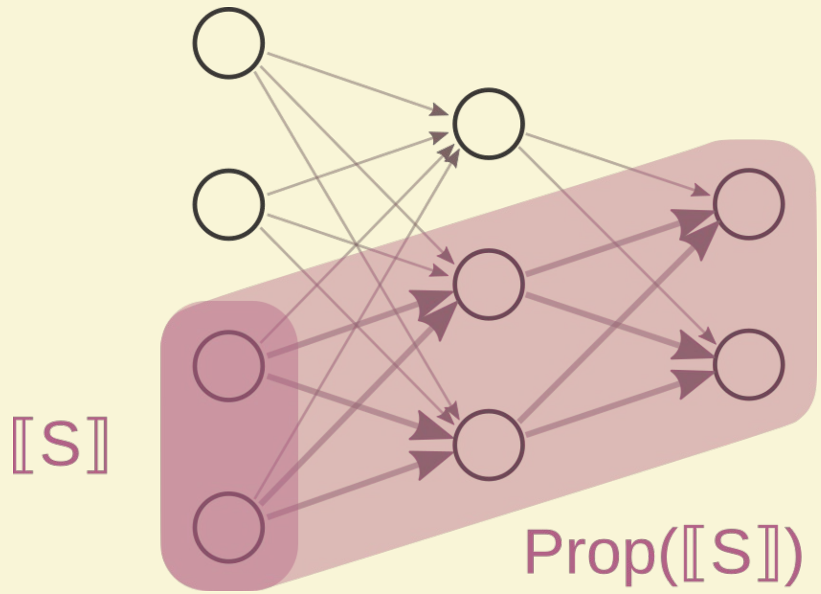
Repeat this update until a fixed point!
i.e. until the weights are "maximally high"

We call the resulting net **Hebb*(N, ⟦S⟧)**

D. Hebb. The Organization of Behavior. Psychology Press, 1949.

Iterated Hebbian Learning

Neurons that fire together wire together

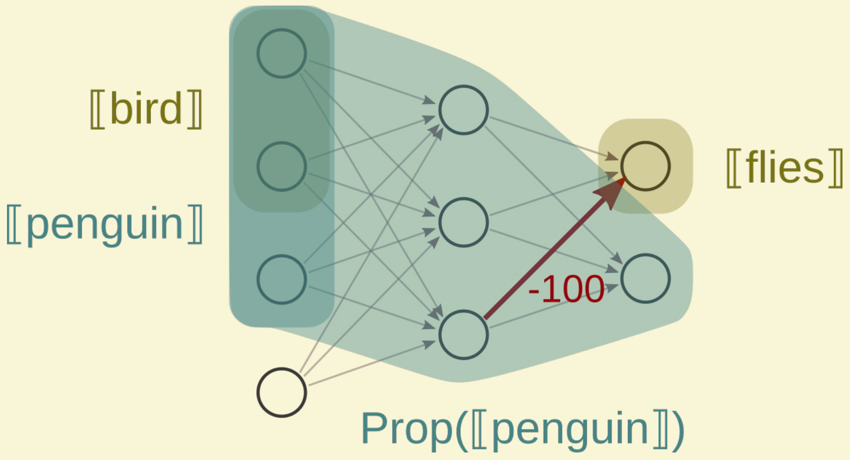


Repeat this update until a fixed point!
i.e. until the weights are "maximally high"

We call the resulting net **Hebb*(N, ⟦S⟧)**

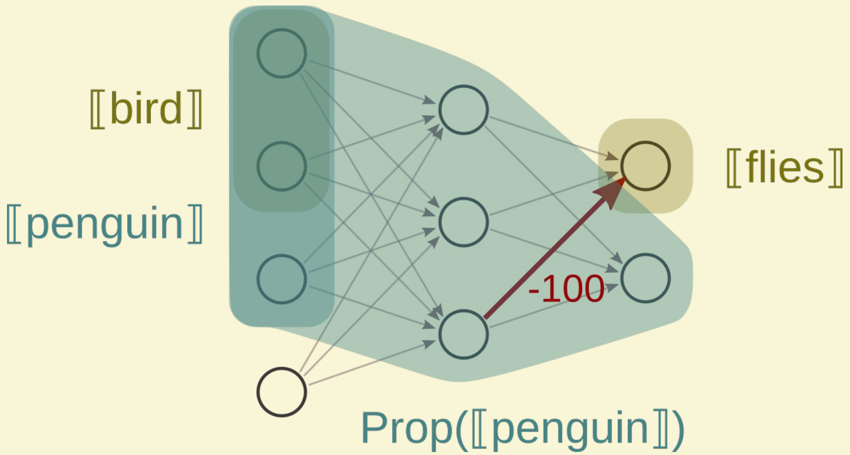
D. Hebb. The Organization of Behavior. Psychology Press, 1949.

Example: Learning Wrecks the Model!



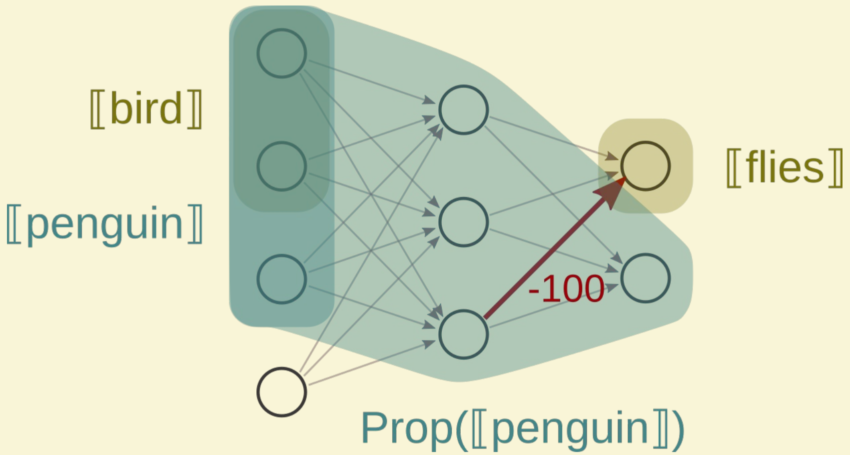
Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

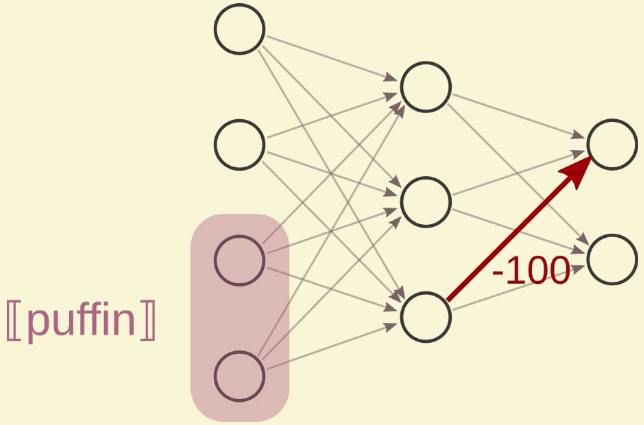
Example: Learning Wrecks the Model!



Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

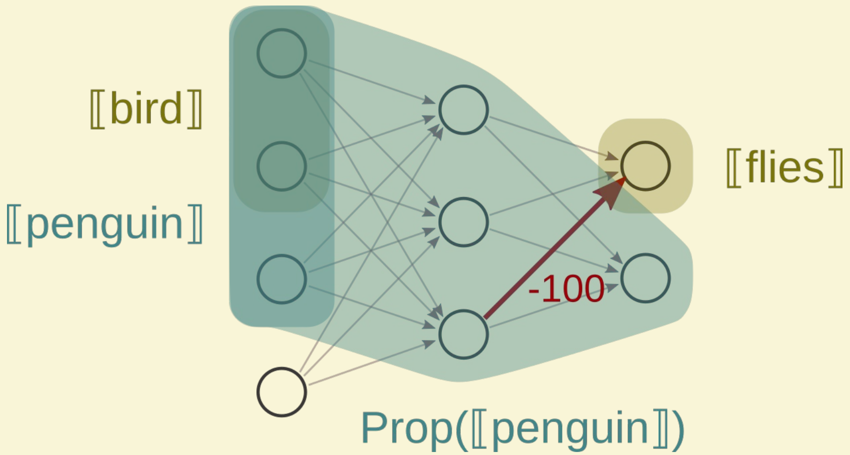
Example: Learning Wrecks the Model!

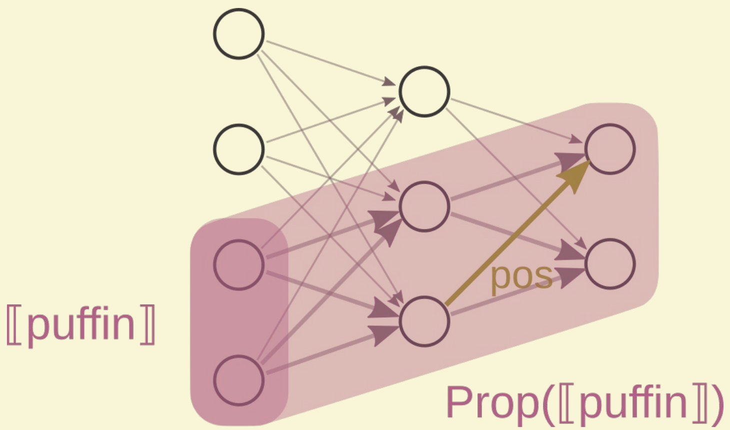




Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

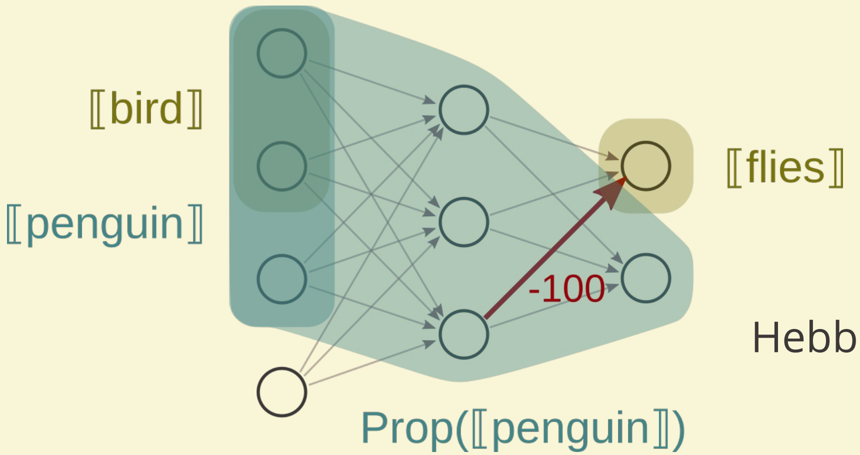
Example: Learning Wrecks the Model!



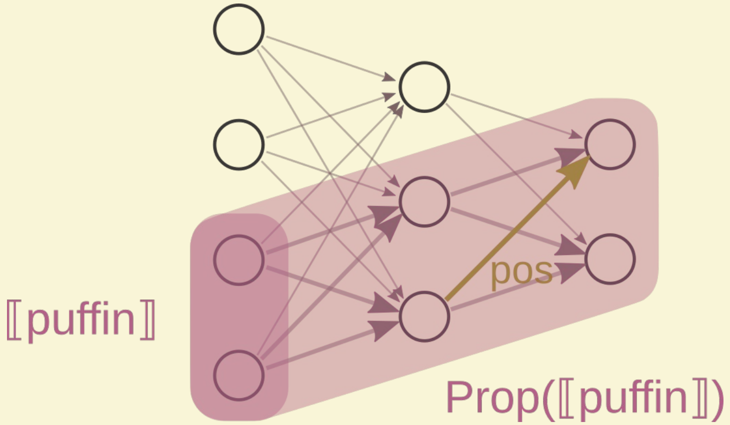


Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

Example: Learning Wrecks the Model!

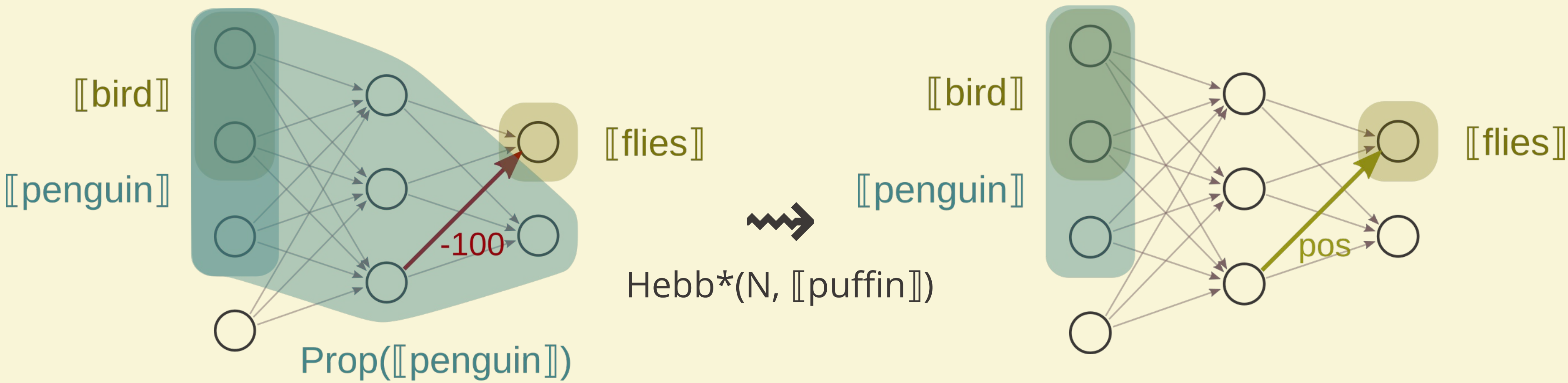


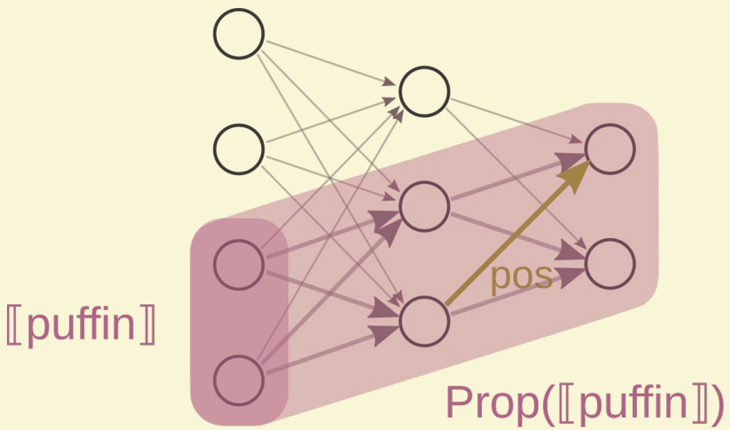
Hebb*(N, ⟦puffin⟧)



Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

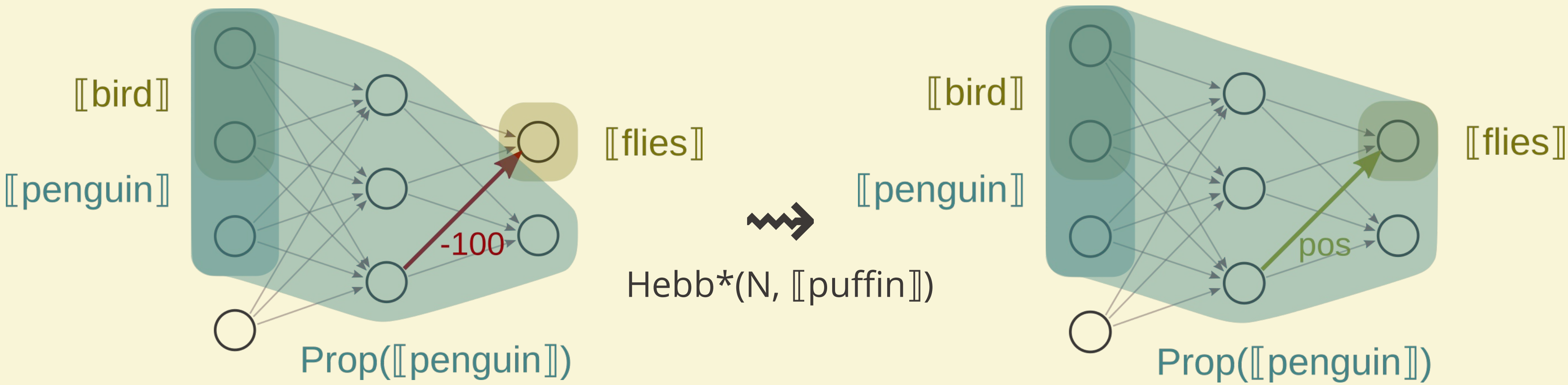
Example: Learning Wrecks the Model!

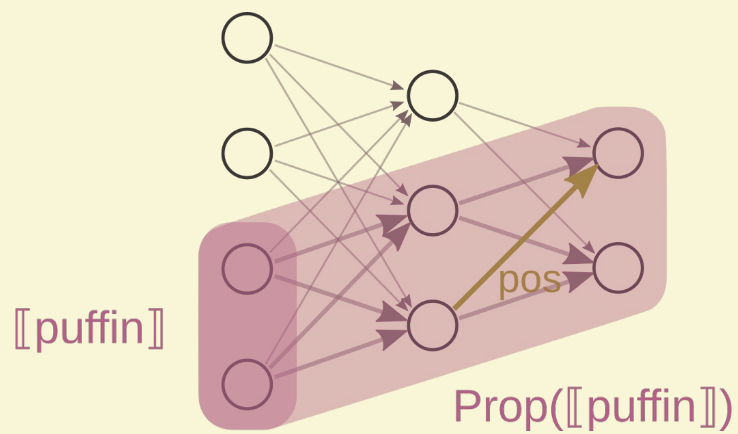




Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

Example: Learning Wrecks the Model!





Louis Agassiz Fuertes. Atlantic Puffin (1932). Watercolor and pencil on paper. From *Portraits of New England Birds*. Commonwealth Of Massachusetts, 1932. Edited by Sabrina Schultz Kisby.

Logic & Formal Semantics

Syntax. We consider the language:

$$A, B \in p \mid \neg A \mid A \wedge B \mid \mathbf{K} A \mid \mathbf{T} A$$

We define the duals $\langle \mathbf{K} \rangle, \langle \mathbf{T} \rangle$ as usual. We can express $A \Rightarrow B$ as $\mathbf{T} A \to B$ ("the typical A is B").

Semantics. We map each formula to a state:

$$[\![p]\!] = V(p) \quad [\![\neg A]\!] = [\![A]\!]^{\complement} \quad [\![A \wedge B]\!] = [\![A]\!] \cap [\![B]\!]$$

$$\begin{aligned} [\![\mathbf{K} A]\!] &= \{n \mid n \text{ is graph-reachable from } A\} \\ [\![\mathbf{T} A]\!] &= \mathrm{Prop}([\![A]\!]) \end{aligned}$$

Definition. $N, w \models A$ iff $w \in [\![A]\!]$

$$[\![[A]B]\!]_N = [\![B]\!]_{\mathrm{Hebb}^*(N, [\![A]\!])}$$

Can we completely characterize $[A]$'s effect on the net?

Kisby, C., Blanco, S., and Moss, L. The Logic of Hebbian Learning. In The International FLAIRS Conference Proceedings, volume 35, 2022.

Main Results

Theorem. The following axioms are sound:

$$\begin{array}{lcll}
[\varphi]p & \leftrightarrow & p & \text{for propositions } p \\
[\varphi]\neg\psi & \leftrightarrow & \neg[\varphi]\psi & \\
[\varphi](\psi \wedge \rho) & \leftrightarrow & [\varphi]\psi \wedge [\varphi]\rho & \\
[\varphi]\mathbf{K}\psi & \leftrightarrow & \mathbf{K}[\varphi]\psi & \\
[\varphi]\mathbf{T}\psi & \leftrightarrow & \mathbf{T}([\varphi]\psi \wedge (\mathbf{T}\varphi \vee \mathbf{K}(\mathbf{T}\varphi \vee \mathbf{T}[\varphi]\psi))) &
\end{array}$$

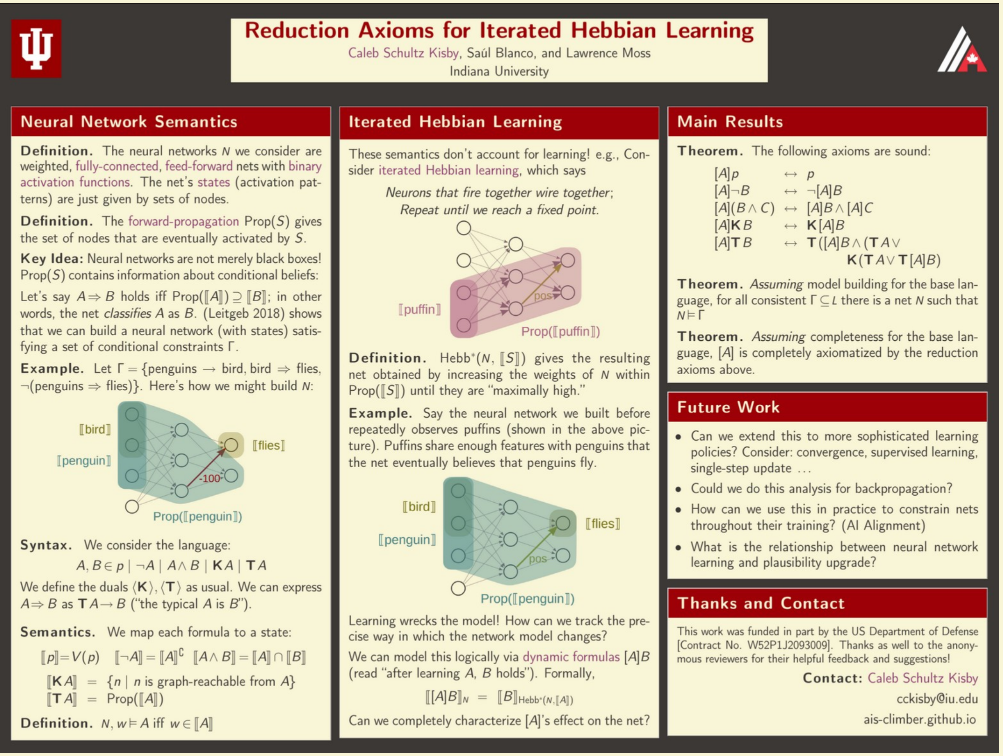
Theorem. **Assuming** model building for the base language: For all consistent $\Gamma \subseteq \mathcal{L}$ there is a net $\mathcal{N}$ such that $\mathcal{N} \models \Gamma$.

Theorem. **Assuming** completeness for the base language: $[\varphi]$ is completely axiomatized by the reduction axioms from before.

Future Work

- **Stabilized Learning Policies.** Hebb* increases the weights until they're maximally large. But stabilized hebbian learning (e.g. Oja's rule) increases weights towards a convergent point.
- **Single-step Update.** We often want guarantees for what the neural network learns *at each step*. This is the heart of AI Alignment.
- **What about Backpropagation?** This is our major long-term goal!

Check Out Our Poster + Paper!



Reduction Axioms for Iterated Hebbian Learning

Caleb Schultz Kisby, Saúl Blanco, and Lawrence Moss
Indiana University

Neural Network Semantics

Definition. The neural networks N we consider are weighted, fully-connected, feed-forward nets with binary activation functions. The net's states (activation patterns) are just given by sets of nodes.

Definition. The forward-propagation $\mathsf{Prop}(S)$ gives the set of nodes that are eventually activated by S.

Key Idea: Neural networks are not merely black boxes! $\mathsf{Prop}(S)$ contains information about conditional beliefs:

Let's say $A \Rightarrow B$ holds iff $\mathsf{Prop}([\![A]\!]) \supseteq [\![B]\!]$; in other words, the net *classifies* A as B. (Leitgeb 2018) shows that we can build a neural network (with states) satisfying a set of conditional constraints Γ.

Example. Let $\Gamma = \{\text{penguins} \to \text{bird}, \text{bird} \Rightarrow \text{flies}, \neg(\text{penguins} \Rightarrow \text{flies})\}$. Here's how we might build N:

Syntax. We consider the language:

$$A, B \in p \mid \neg A \mid A \wedge B \mid \mathbf{K}A \mid \mathbf{T}A$$

We define the duals $\langle \mathbf{K} \rangle, \langle \mathbf{T} \rangle$ as usual. We can express $A \Rightarrow B$ as $\mathbf{T}A \to B$ ("the typical A is B").

Semantics. We map each formula to a state:

$$[\![p]\!] = V(p) \quad [\![\neg A]\!] = [\![A]\!]^{\complement} \quad [\![A \wedge B]\!] = [\![A]\!] \cap [\![B]\!]$$

$$[\![\mathbf{K}A]\!] = \{n \mid n \text{ is graph-reachable from } A\}$$

$$[\![\mathbf{T}A]\!] = \mathsf{Prop}([\![A]\!])$$

Definition. $N, w \models A$ iff $w \in [\![A]\!]$

Iterated Hebbian Learning

These semantics don't account for learning! e.g., Consider iterated Hebbian learning, which says

Neurons that fire together wire together;
Repeat until we reach a fixed point.

Definition. $\mathsf{Hebb}^*(N, [\![S]\!])$ gives the resulting net obtained by increasing the weights of N within $\mathsf{Prop}([\![S]\!])$ until they are "maximally high."

Example. Say the neural network we built before repeatedly observes puffins (shown in the above picture). Puffins share enough features with penguins that the net eventually believes that penguins fly.

Learning wrecks the model! How can we track the precise way in which the network model changes?

We can model this logically via dynamic formulas $[A]B$ (read "after learning A, B holds"). Formally,

$$[\![[A]B]\!]_N = [\![B]\!]_{\mathsf{Hebb}^*(N, [\![A]\!])}$$

Can we completely characterize $[A]$'s effect on the net?

Main Results

Theorem. The following axioms are sound:

$$\begin{array}{lcl} [A]p & \leftrightarrow & p \\ [A]\neg B & \leftrightarrow & \neg[A]B \\ [A](B \wedge C) & \leftrightarrow & [A]B \wedge [A]C \\ [A]\mathbf{K}B & \leftrightarrow & \mathbf{K}[A]B \\ [A]\mathbf{T}B & \leftrightarrow & \mathbf{T}([A]B \wedge (\mathbf{T}A \vee \\ & & \quad \mathbf{K}(\mathbf{T}A \vee \mathbf{T}[A]B) \end{array}$$

Theorem. *Assuming* model building for the base language, for all consistent $\Gamma \subseteq L$ there is a net N such that $N \models \Gamma$

Theorem. *Assuming* completeness for the base language, $[A]$ is completely axiomatized by the reduction axioms above.

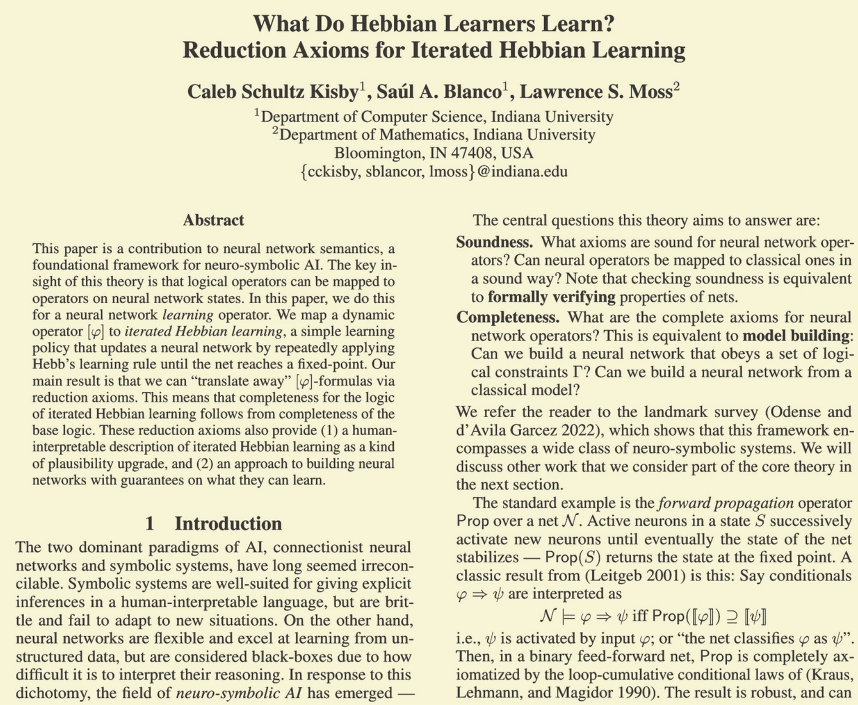
Future Work

- Can we extend this to more sophisticated learning policies? Consider: convergence, supervised learning, single-step update …
- Could we do this analysis for backpropagation?
- How can we use this in practice to constrain nets throughout their training? (AI Alignment)
- What is the relationship between neural network learning and plausibility upgrade?

Thanks and Contact

This work was funded in part by the US Department of Defense [Contract No. W52P1J2093009]. Thanks as well to the anonymous reviewers for their helpful feedback and suggestions!

Contact: Caleb Schultz Kisby
cckisby@iu.edu
ais-climber.github.io



What Do Hebbian Learners Learn?
Reduction Axioms for Iterated Hebbian Learning

Caleb Schultz Kisby[1], **Saúl A. Blanco**[1], **Lawrence S. Moss**[2]
[1]Department of Computer Science, Indiana University
[2]Department of Mathematics, Indiana University
Bloomington, IN 47408, USA
{cckisby, sblancor, lmoss}@indiana.edu

Abstract

This paper is a contribution to neural network semantics, a foundational framework for neuro-symbolic AI. The key insight of this theory is that logical operators can be mapped to operators on neural network states. In this paper, we do this for a neural network *learning* operator. We map a dynamic operator $[\varphi]$ to *iterated Hebbian learning*, a simple learning policy that updates a neural network by repeatedly applying Hebb's learning rule until the net reaches a fixed-point. Our main result is that we can "translate away" $[\varphi]$-formulas via reduction axioms. This means that completeness for the logic of iterated Hebbian learning follows from completeness of the base logic. These reduction axioms also provide (1) a human-interpretable description of iterated Hebbian learning as a kind of plausibility upgrade, and (2) an approach to building neural networks with guarantees on what they can learn.

1 Introduction

The two dominant paradigms of AI, connectionist neural networks and symbolic systems, have long seemed irreconcilable. Symbolic systems are well-suited for giving explicit inferences in a human-interpretable language, but are brittle and fail to adapt to new situations. On the other hand, neural networks are flexible and excel at learning from unstructured data, but are considered black-boxes due to how difficult it is to interpret their reasoning. In response to this dichotomy, the field of *neuro-symbolic AI* has emerged —

The central questions this theory aims to answer are:

Soundness. What axioms are sound for neural network operators? Can neural operators be mapped to classical ones in a sound way? Note that checking soundness is equivalent to **formally verifying** properties of nets.

Completeness. What are the complete axioms for neural network operators? This is equivalent to **model building**: Can we build a neural network that obeys a set of logical constraints Γ? Can we build a neural network from a classical model?

We refer the reader to the landmark survey (Odense and d'Avila Garcez 2022), which shows that this framework encompasses a wide class of neuro-symbolic systems. We will discuss other work that we consider part of the core theory in the next section.

The standard example is the *forward propagation* operator Prop over a net $\mathcal{N}$. Active neurons in a state S successively activate new neurons until eventually the state of the net stabilizes — $\mathsf{Prop}(S)$ returns the state at the fixed point. A classical result from (Leitgeb 2001) is this: Say conditionals $\varphi \Rightarrow \psi$ are interpreted as

$$\mathcal{N} \models \varphi \Rightarrow \psi \text{ iff } \mathsf{Prop}([\![\varphi]\!]) \supseteq [\![\psi]\!]$$

i.e., ψ is activated by input φ; or "the net classifies φ as ψ". Then, in a binary feed-forward net, Prop is completely axiomatized by the loop-cumulative conditional laws of (Kraus, Lehmann, and Magidor 1990). The result is robust, and can

Contact:
Caleb Schultz Kisby
cckisby@iu.edu
https://ais-climber.github.io/